



Decomposition algorithm for large-scale two-stage unit-commitment

Wim van Ackooij, Jérôme Malick

► To cite this version:

Wim van Ackooij, Jérôme Malick. Decomposition algorithm for large-scale two-stage unit-commitment. *Annals of Operations Research*, 2016, 238 (1), pp.587-613. 10.1007/s10479-015-2029-8 . hal-01145653

HAL Id: hal-01145653

<https://hal.science/hal-01145653>

Submitted on 24 Apr 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Decomposition algorithm for large-scale two-stage unit-commitment

W. van Ackooij · J. Malick

Received: date / Accepted: date

Abstract Everyday, electricity generation companies submit a generation schedule to the grid operator for the coming day; computing an optimal schedule is called the unit-commitment problem. Generation companies can also occasionally submit changes to the schedule, that can be seen as intra-daily incomplete recourse actions. In this paper, we propose a two-stage formulation of unit-commitment, wherein both the first and second stage problems are full unit-commitment problems. We present a primal-dual decomposition approach to tackle large-scale instances of these two-stage problems. The algorithm makes extensive use of warm-started bundle algorithms, and requires no specific knowledge of the underlying technical constraints. We provide an analysis of the theoretical properties of the algorithm, as well as computational experiments showing the interest of the approach for real-life large-scale unit-commitment instances.

Keywords two-stage integer programming, stochastic unit-commitment, price decomposition, convex duality, nonsmooth optimization, bundle methods

Mathematics Subject Classification (2000) 49M37 · 65K05 · 90C15

1 Introduction

In energy management, a key problem known as “unit-commitment” deals with finding a minimal cost production schedule that satisfies the operational constraints of the production units and that meets customer load as closely as possible. (see the recent review [47]). Since operational constraints involve delays (start-up delays, etc...), the computed production schedule is often determined quite ahead of real-time. In electrical systems wherein renewable generation has overall high generation capacity, uncertainty is strongly present and has a key impact on both “feasibility” and “optimality” of the executed production schedule. In practice, spinning reserves and intra-daily changes to the schedule allow the operator to partially account for uncertainty. Highly binding operational constraints might give rise to difficult situations, wherein the quest for “feasibility” induces a heavy cost. As such, computing a schedule having seen at least part of the uncertainty might turn out to be less costly eventually. Ideally, two-stage unit-commitment problems would consider uncertainty on customer load, on renewable generation, on inflows for the hydro reservoirs and on unit availability. In this paper, we consider here the first two sources of uncertainty. The third one could also be integrated in our approach, following, e.g., [52].

Wim van Ackooij
EDF R&D. OSIRIS 1, avenue du Général de Gaulle, F-92141 Clamart France
E-mail: wim.van-ackooij@edf.fr

Jérôme Malick
CNRS, LJK, F-38000 Grenoble, France
E-mail: jerome.malick@inria.fr

Stochastic unit-commitment models are less common in the literature than deterministic ones, and none of them could capture the situation of this paper. Many existing approaches (including [38, 40, 49, 50, 54]) use scenario trees where uncertainty in each node is known when the decision is taken. The robust unit-commitment models [4, 57] decouple commitment decisions (starting/stopping status of each unit) from dispatch decisions (power output), which are taken only when uncertainty is known. In these approaches, it is unclear what schedule (including power output) has to be sent to the grid operator – which is our practical motivation. The two-stage model of [58] allows to adapt some commitment decisions and the resulting stochastic unit-commitment problems are amenable to mixed-integer linear programming solvers by using a technique of [45]. However the problems tackled (with only 5 thermal units) is an order of magnitude smaller than the problems we target.

In this paper, we formalize the situation as two-stage models, wherein both the first and second stage are full unit-commitment problems. We see the occasional changes of schedule as intra-daily recourse actions, incomplete because of technical constraints on generation. We allow a rich modelling of these operation constraints (possibly, non-linear, non-convex, with discrete variables) that lead to large-scale mixed-integer problems, out of reach for direct approaches and even for existing decomposition methods. We propose a tractable primal-dual decomposition approach for these large-scale unit-commitment problems, attacking both stages by duality. Our algorithm uses the same ingredients (deterministic subproblems, cutting plane models, bundle methods, primal recovery heuristics) as in the deterministic case (see e.g. [18, 20, 59]). We pay a special attention to hot-starting which is a critical issue in view of tackling large-scale problems.

Here is the outline of this paper. First, section 2 introduces unit-commitment problems in the deterministic and the stochastic cases: notation and assumptions are presented. The state-of-the art is sketched. Section 3 presents the decomposition algorithm, relaxing coupling constraints in both the first and second stages. The method is put into perspective in section 4, where the convexification effect, the interpretation of stopping tests, and the convergence properties are analyzed. In the final section, we present numerical illustrations on real-life unit-commitment instances.

2 Stochastic unit-commitment: notation, assumptions, and state of the art

2.1 A structural viewpoint on deterministic unit-commitment: notation and presentation

Unit-commitment problems are already challenging in a deterministic setting: the units are coupled through constraints such as the offer-demand equilibrium constraint, and are subject to many technical constraints, specific to their type (thermal, hydraulic, contracts). We consider here m units indexed by $i = 1, \dots, m$. We denote the decision variables (including production) of unit i by $x_i \in \mathbb{R}^{n_i}$, its production cost by $f_i(x_i)$ and its specific production constraints by $x_i \in X_i$. The decision variable is thus $x = (x_1, \dots, x_m) \in \mathbb{R}^n$ where $\sum_{i=1}^m n_i = n$. Units are linked through the offer-demand equilibrium constraints, that state that deviation between production and customer load has to remain small. These constraints have the typical form

$$s^d \leq D - Ax \leq s^u, \quad (1)$$

where $s^d, s^u \in \mathbb{R}^T$ are operator chosen bounds, T is the number of time steps in the considered time horizon, $D \in \mathbb{R}^T$ is the customer load, and A the $T \times n$ matrix summing up the decision vector $x = (x_1, \dots, x_m) \in \mathbb{R}^n$. An abstract formulation of deterministic unit-commitment has then the following form

$$\begin{aligned} \min_{x=(x_1, \dots, x_m)} \quad & \sum_{i=1}^m f_i(x_i), \\ \text{s.t.} \quad & x_i \in X_i \subseteq \mathbb{R}^{n_i}, i = 1, \dots, m \\ & s^d \leq D - Ax \leq s^u. \end{aligned}$$

Using aggregated objects $f(x) = \sum_{i=1}^m f_i(x_i)$,

$$X^1 := \prod_{i=1}^m X_i \quad \text{and} \quad X^2 := \left\{ x \in \mathbb{R}^n : s^d \leq D - Ax \leq s^u \right\}$$

we can write the above unit-commitment problem in a short manner as:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{s.t.} \quad & x \in X^1 \cap X^2. \end{aligned} \quad (2)$$

Practical formulations of (2) often lead to mixed-integer problems. Since there now exist strong commercial solvers, this has become the major approach for solving unit-commitment (e.g., [12, 35, 36]).

However the success of this direct approach strongly hinges on the the modelling detail that we decide to integrate in the subproblems. The sets X_i can indeed require a large number of modelling variables together with nonlinear terms in the objective and constraints. For example, hydraulic “units” are typically entire hydro valleys operating independently. Key constraints are bilateral bounds on volume in each reservoir, flow constraints, and technical constraints on turbinning/pumping operation. Moreover, the value of water is typically computed by a mid-term planning tool. Uncertainty on inflows can also be taken into account, for instance by using joint probabilistic constraints as in [52]. In quite a similar way, thermal units are subject to many technical constraints on power variations, starts, ramping rates, minimum up/down times... Most of these constraints imply non-convexities and typical modelling involves binary variables. Therefore, optimizing a single unit with respect to a price signal can already be quite challenging, see, e.g., [22] for hydro valley units and [23, 30] for thermal units.

In large-scale systems, or in systems requiring substantial modelling detail, decomposition approaches for (2) appear as the only viable solution. The unit-commitment instances for the French system for example are both large scale and require substantial modelling detail. In order to tackle such problems, the coupling constraints are often dualized, using Lagrangian techniques (see [18, 25] and references therein). An important feature of this Lagrangian decomposition approach is that it provides marginal prices that are useful for the operator of the system. Though this approach leads to non-feasible primary solutions, it also gives good starting points for primal recovering heuristics (see [3, 18, 24, 44, 53, 59]).

2.2 Recourse in unit-commitment, assumptions

A solution of problem (2) defines a production schedule x (commitment decisions and power output), sent to the grid-operator before being activated and before observing uncertainty. In real time, a new production schedule, redefining both commitment decisions and power output, can be sent to the grid-operator at specific moments in time. This implies that the recourse problem is exactly of the same structure as (2) and has the same complexity, but with a smaller time horizon.

More precisely, we consider an abstract random process $\xi \in \mathbb{R}^k$ affecting uncertainty on customer load and renewable generation. Observing this process at time step $\tau \in \{1, \dots, T\}$ results in “observing” the net customer load $D(\xi) \in \mathbb{R}^T$. This load consists of $D(\xi)_1, \dots, D(\xi)_\tau$, the actually observed net customer load of the previous time $t = 1, \dots, \tau$ and $D(\xi)_{\tau+1}, \dots, D(\xi)_T$, the current best forecast of net customer load after τ . We introduce the appropriate modification of X^2 involving the change in D denoted

$$X^2(\xi) := \left\{ y \in \mathbb{R}^n : s^d \leq D(\xi) - Ay \leq s^u \right\},$$

and the recourse cost function $c : \mathbb{R}^n \times \mathbb{R}^k \rightarrow \mathbb{R} \cup \{+\infty\}$ as:

$$c(x, \xi) := \begin{cases} \min_{y \in \mathbb{R}^n} & f(y) \\ \text{s.t.} & y \in X^1 \cap X^2(\xi), \\ & Px = Py, \end{cases} \quad (3)$$

where P is a $\ell \times n$ matrix having a single non-zero element for each line and column. The equation $Px = Py$ models the fact that the power output of each unit prior to τ is fixed and that the recourse decision y can only modify power output after τ . The segment of y corresponding to decisions taken prior to time τ can be seen as duplicated according to the scenarios. The constraint $Px = Py$ can thus be seen as a non-anticipativity constraint (see, e.g., [10]) since it enforces that the decisions are equal on all scenarios.

To simplify presentation, we consider in this paper that the process ξ has a discrete distribution: its realizations (called scenarios) are labelled

$$\Xi := \{\xi_1, \dots, \xi_S\} \quad \text{with associated probabilities } p_1, \dots, p_S. \quad (4)$$

We refer to [21] for a recent work on load scenario forecasting. The expected recourse cost function is then naturally defined as

$$v : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\} \quad v(x) := \mathbb{E}(c(x, \xi)) = \sum_{j=1}^S p_j c(x, \xi_j).$$

This leads to the following formulation of the two-stage unit-commitment problem, which is the problem we focus on in this paper

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) + v(x) \\ \text{s.t.} \quad & x \in X^1 \cap X^2. \end{aligned} \quad (5)$$

The constraints of the problem (5) are the same as the initial unit-commitment problem (2). As explained in the previous section, X^1 can contain many binary variables, implicit constraints, joint probabilistic constraints... In this paper, we do not suppose to know X^1 explicitly; we just make the following assumptions on our problem (5):

- Practical assumption 1: we can solve (approximatively) the (sub)problems defined as minimizing the sum of f_i and a linear term over X_i :

$$\min_{x_i \in X_i} f_i(x_i) + b_i^\top x_i.$$

- Practical assumption 2: Lagrangian based primal recovery heuristics (e.g., [7, 20, 48]) are readily available to build a primal feasible solution out of primal iterates and dual information.
- Theoretical assumption on X^1 : each $X_i \subset \mathbb{R}^{n_i}$ is compact. The compactness of X^1 implies that its convex hull $\text{conv}(X^1)$ is compact as well (by [28, III.1.4.3]). Thus the sets $X^1 \cap X^2$ and $\text{conv}(X^1) \cap X^2$, and $\text{conv}(X^1 \cap X^2)$, that will appear in our analysis, are also compact.
- Theoretical assumption on f : each $f_i : \mathbb{R}^{n_i} \rightarrow \mathbb{R}$ is a closed convex function on \mathbb{R}^{n_i} . In view of the first practical assumption above, f_i should be simple, as piece-wise linear or quadratic. We also assume the f_i are bounded from below; without loss of generality, we consider $f_i \geq 0$.
- Consistency assumption: observe that (5) has implicit constraints coming from v , whose domain is

$$\text{dom}(v) := \{x : v(x) < +\infty\} = \left\{x : \text{for all } \xi \in \Xi, \exists y \in X^2(\xi) \text{ such that } Px = Py\right\}.$$

Our final assumption is that $\text{dom}(v)$ is nonempty, so that there exists a solution to (5).

2.3 Limitations of existing decomposition approaches

Two-stage unit-commitment problems are very difficult in a large-scale setting. In fact, computing $c(x, \xi)$ for a fixed (x, ξ) is a full unit-commitment problem, which is already difficult when the set X^1 is complex and m large. Solving our problem (5) therefore requires a decomposition approach, that can be either primal (Benders decomposition) or dual (Lagrangian decomposition). Let us sketch these approaches and the existing methods proposed for related stochastic unit-commitment problems.

In a primal approach to (5), the recourse cost function at given x would involve modifications of X^1 with initial condition x and constraints related to time steps after τ . Making appropriate changes to handle modified constraint sets might involve significant additional modelling, that we want to avoid. Note that, for specific two-stage models [50] proposes a Benders decomposition approach plugging the two-stage cost function into the first stage. This primal approach considers a simple form for the second-stage problem (which is fuel requirement problem).

A dual approach to (5), by duplicating variables along scenarios and dualizing the non-anticipativity conditions, would get rid of x . Hence making unclear how to restore feasibility (i.e., $x \in X^1$ and $Px = Py$). The strategy of [10], that embeds the Lagrangian scheme within a branch and bound method, is not possible in our setting, where the deterministic model is already too large to fit in such a scheme. A dual decomposition is considered in [49] for a specific stochastic unit-commitment problem. The commitment decisions are the only binary variables in X^1 and a progressive hedging algorithm [41] is used to average these out. The sets X^1 are still rather simple and one can easily fix the obtained schedule to create a X^1 -feasible solution. This would not be possible when X^1 is defined by many binary variables (e.g., some realistic thermal sub-problems might require up to 100 binary variables per time step). The recent work [13] also presents a dedicated progressive hedging algorithm.

Another Lagrangian-based decomposition is proposed by [11] where uncertainty is discretized on a scenario-tree, and an augmented Lagrangian dual is used to relax the coupling constraints of $X^2(\xi)$. The subproblems are then stochastic 1-unit optimization problems, requiring important modifications of X^1 and special resolution approaches, which we want to avoid here. Similar approaches [16, 27, 37, 38] can be viewed as geographical decomposition, following the terminology of [17]. These methods all lead to stochastic subproblems requiring special treatments (by dynamic programming, for example as in [38]). Finally, in [39], integer variables remain present in the second stage after decomposition of uncertain system wide constraints by stochastic Lagrange multipliers.

To sum up, all the existing primal or dual approaches

- either involve significant simplifications of the second stage,
- or make unclear how to recover feasible first stage solutions $x \in X^1$,
- or make significant changes to the set of technical constraints X^1 .

None of the existing decomposition approaches could tackle our stochastic unit-commitment problem (5) in our targeted applications. In the next section, we propose a primal-dual decomposition combining good aspects of both primal and dual approaches.

3 Primal-dual decomposition approach to two-stage unit-commitment

This section presents our decomposition algorithm for solving the large-scale stochastic unit-commitment problem (5). Our approach is primal-dual as it uses relaxation for both stages: the second stage dual algorithm constructs linearizations of the objective function in a way that we can store information for later use (section 3.1), and the first stage dual algorithm produces lower bounds on the optimal value (section 3.2). We introduce notation and recall basic properties in the first two sections, then we present our algorithm in section 3.3.

3.1 Dual approach to the 2nd stage: linearizations of the objective

We apply here the standard Lagrangian relaxation mechanism (see e.g. [31]) to the second stage problems (3) to build a cutting planes model of the recourse function v . For a given $(x, \xi) \in \mathbb{R}^n \times \mathbb{R}^k$, we relax the coupling constraints in $X^2(\xi)$ and the linking constraints $Px = Py$. For any variable $(\lambda_1, \lambda_2, \lambda_3)$ in the dual space $\Lambda := \mathbb{R}^\ell \times \mathbb{R}_+^T \times \mathbb{R}_+^T$, the dual function has the following structure:

$$\theta_{x,\xi}(\lambda_1, \lambda_2, \lambda_3) = \lambda_1^\top Px + \lambda_2^\top (s^d - D(\xi)) + \lambda_3^\top (D(\xi) - s^u) + \bar{\theta}(\lambda_1, \lambda_2, \lambda_3), \quad (6)$$

where the function $\bar{\theta}$, independent from x and ξ , is defined by

$$\bar{\theta}(\lambda_1, \lambda_2, \lambda_3) := \min_{y \in X^1} f(y) - \lambda_1^\top Py + (\lambda_2 - \lambda_3)^\top Ay. \quad (7)$$

The dual function $\theta_{x,\xi}$ is concave by construction and computing a value and a subgradient amounts to computing $\bar{\theta}(\lambda_1, \lambda_2, \lambda_3)$ and $g \in \partial \bar{\theta}(\lambda_1, \lambda_2, \lambda_3)$. The interest of the Lagrangian approach resides in the fact

that this computation decomposes over the production units:

$$\bar{\theta}(\lambda_1, \lambda_2, \lambda_3) = \sum_{i=1}^m \min_{y_i \in X_i} f_i(y_i) + y_i^\top (A^\top(\lambda_2 - \lambda_3) - P^\top \lambda_1)_i.$$

Bundle algorithms (see e.g. [29, Chap.XV]) are the methods of choice for maximizing the dual function. We emphasize that all the information on $\bar{\theta}$ computed during maximizing $\theta_{x,\xi}$ can be stored and used to warmstart the maximization of $\theta_{x',\xi'}$ for another (x', ξ') . This will be of importance for the numerical efficiency of the method.

It is well-known that maximizing the dual function provides information on the primal value function c . Specifically, by weak duality, we have for (x, ξ) and any $(\lambda_1, \lambda_2, \lambda_3) \in \Lambda$,

$$c(x, \xi) \geq \lambda_1^\top P x + \lambda_2^\top (s^d - D(\xi)) + \lambda_3^\top (D(\xi) - s^u) + \bar{\theta}(\lambda_1, \lambda_2, \lambda_3).$$

For a fixed (\bar{x}, ξ) , this yields

$$\begin{aligned} c(x, \xi) &\geq \lambda_1^\top P(x - \bar{x}) + \lambda_1^\top P\bar{x} + \lambda_2^\top (\bar{x}, \xi_j)^\top (s^d - D(\xi)) + \lambda_3^\top (\bar{x}, \xi_j)^\top (D(\xi) - s^u) + \bar{\theta}(\lambda_1, \lambda_2, \lambda_3) \\ &\geq \lambda_1^\top P(x - \bar{x}) + \theta_{\bar{x}, \xi}(\lambda_1, \lambda_2, \lambda_3). \end{aligned}$$

We thus have a linearization at \bar{x} for the mapping $c(\cdot, \xi)$ and, obviously, the best linearizations are those given by the optimal dual solutions for (\bar{x}, ξ) . By integration over all the scenarios, we directly get a linearization of the recourse function v . Repeating this for several points x^1, \dots, x^{k-1} allows us to define a cutting-plane model of v

$$\tilde{v}_k(x) := \max_{i=1, \dots, k-1} \left\{ (\bar{g}^i)^\top (x - x^i) + \bar{v}^i \right\} \leq v(x), \quad (8)$$

with appropriately aggregated \bar{g}^i and \bar{v}^i . More specifically, by (4), we have

$$\bar{g}^i := P^\top \left(\sum_{j=1}^S p_j \lambda_1(x^i, \xi_j) \right) \quad \text{and} \quad \bar{v}^i := \sum_{j=1}^S p_j \theta_{x^i, \xi_j} \left(\lambda_1(x^i, \xi_j), \lambda_2(x^i, \xi_j), \lambda_3(x^i, \xi_j) \right) \quad (9)$$

for the dual variables $(\lambda_1(x^i, \xi_j), \lambda_2(x^i, \xi_j), \lambda_3(x^i, \xi_j))$. Note that we aggregate linearizations as above to simplify presentation. As usual in stochastic programming (see e.g. [43]), the cuts could also be combined in other ways; among them multi-cuts of [5, 6] or the partially aggregating cuts of [55]. Our algorithm is compatible with any such versions; the effect of multi-cuts is illustrated in section 5.2.

3.2 Dual approach to the first stage problem: lower bound for our problem

We use now the cutting plane model (8) to get a lower bound for our problem (5). For a fixed k , we consider the following approximated first stage optimization problem, wherein v is replaced by \tilde{v}_k

$$\left\{ \begin{array}{ll} \min_{x \in X^1 \cap X^2} f(x) + \tilde{v}_k(x), & \text{written as} \end{array} \right. \left\{ \begin{array}{ll} \min_{(x, \nu)} & f(x) + \nu, \\ \text{s.t.} & (\bar{g}^i)^\top (x - x_i) + \bar{v}^i \leq \nu, \quad i = 1, \dots, k-1 \\ & x \in X^1 \cap X^2. \end{array} \right. \quad (10)$$

We dualize all the coupling constraints: those in X^2 and those provided by \tilde{v}_k . Gathering the linearizations in $G_k := (\bar{g}^i)_{i=1, \dots, k-1} \in \mathbb{R}^{n \times (k-1)}$ and $b_k := (\bar{v}^i)_{i=1, \dots, k-1} \in \mathbb{R}^k$, the concave dual function of (10) writes

$$\begin{aligned} \Theta_k(\mu, \nu_1, \nu_2) &= \mu^\top b_k + \nu_1^\top (s^d - D) + \nu_2^\top (D - s^u) \\ &\quad + \sum_{i=1}^m \min_{x_i \in X_i} f_i(x_i) + x_i^\top \left(G_k \mu + A^\top(\nu_1 - \nu_2) \right)_i \end{aligned} \quad (11)$$

for any dual variables $(\mu, \nu_1, \nu_2) \in \mathbb{R}_+^{k-1} \times \mathbb{R}_+^T \times \mathbb{R}_+^T$ with $\sum_{k=1}^k \mu_k = 1$. By weak duality and the fact that $\tilde{v}_k(x) \leq v(x)$, we have that $\Theta_k(\mu, \nu_1, \nu_2)$ is a lower bound on the optimal value of (5). Note that the lower bound thus comes out as the addition of two gaps: the duality gap between (10) and its dual on top of the approximation gap coming from replacing v by \tilde{v}_k . We need to control these two gaps in the algorithm.

3.3 Description of the algorithm

The material of the previous sections allows us to design the following decomposition approach for the two-stage unit-commitment problem (5).

Step 0 (Initialization) Choose the stopping tolerance $\delta_{\text{tol}} > 0$ and a feasibility detection target $\theta_{\text{feas}} > 0$ (strictly greater than the optimal value). Set parameters for the first and second stage bundle algorithms. Choose $(\nu_1^0, \nu_2^0) \in \mathbb{R}_+^T \times \mathbb{R}_+^T$, and set $k = 1$.

Step 1 (First stage) At iteration k , use a bundle method to maximize Θ_k in (11) starting at the current dual variables $(\mu^{k-1}, \nu_1^{k-1}, \nu_2^{k-1})$. Run this algorithm until convergence or just for a few steps; when it is stopped, it returns new dual variables $(\mu^k, \nu_1^k, \nu_2^k)$ such that

$$\Theta_k(\mu^{k-1}, \nu_1^{k-1}, \nu_2^{k-1}) \leq \Theta_k(\mu^k, \nu_1^k, \nu_2^k) \leq \text{optimal value of (5)}.$$

Step 2 (Lagrangian heuristics) Use any heuristic for recovering $x^k \in X^1 \cap X^2$ a primal feasible solution of (10). Define the observed duality gap by $\Delta_G^k := f(x^k) + \check{v}_k(x^k) - \Theta_k(\mu^k, \nu_1^k, \nu_2^k)$.

Step 3 (Second stage, model enrichment) For each couple (x^k, ξ) with $\xi \in \Xi$, maximize $\theta_{x^k, \xi}$ using a hot-started bundle algorithm, stopped if $\theta_{x^k, \xi}(\lambda_1^k, \lambda_2^k, \lambda_3^k) \geq \theta_{\text{feas}}$. Once all second stage problems are processed, add a linearization to the cutting plane model to create \check{v}_{k+1} . Define the approximation error $\Delta_A^k := \check{v}_{k+1}(x^k) - \check{v}_k(x^k)$.

Step 4 (Stopping Test) If $\check{v}_{k+1}(x^k) \leq \theta_{\text{feas}}$, check if $\Delta_A^k < \delta_{\text{tol}}$ holds, in which case, move to the next step. If not, increment k to $k + 1$ and return to Step 1.

Final step (Optional post-treatment) Try to improve x^k by using some (costly) heuristic approaches.

This algorithm has the same components as decomposition algorithms for deterministic unit-commitment, namely cutting plane models, bundle methods and primal recovery heuristics (see e.g. the section on Lagrangian decomposition in [47] and references therein). Let us discuss some points.

First stage problem and Lagrangian heuristics. The dual bundle algorithm of Steps 1 provides good lower bounds together with information (primal-dual iterates defining the so-called pseudo-schedule via the optimality conditions) useful for the primal heuristics of Step 2 constructing near optimal solutions of (10), see [7, 20, 48]). Contrary to the rest of the algorithm, some of these heuristics may use explicit knowledge of X^1 . When the problem is large-scale, the duality gap Δ_G^k obtained by these heuristics can be lower than 0.5% (see e.g., [25]). Notice that, by definition of Δ_G^k , we can write for any $x \in X^1 \cap X^2$

$$f(x) + \check{v}_k(x) \geq \Theta_k(\mu^k, \nu_1^k, \nu_2^k) = f(x^k) + \check{v}_k(x^k) - \Delta_G^k. \quad (12)$$

This shows that, for any k , the iterate x^k is a Δ_G^k -solution of the k -th approximation problem (10).

Model enrichment. In step 3, we enrich the model (8) by adding the linearization given by (9) obtained by solving the S dual second stage problems. As the linearization might not be tight, we call it a “suboptimality cut”. In fact, these suboptimality cuts play a double role of being both optimality and feasibility cuts simultaneously. This will be illustrated in the next section and further studied in section 4.1.

Recall that $\bar{\theta}$ does not depend on (x^k, ξ_j) , so that we can keep the known linearizations of $\bar{\theta}$ from one point to another, and from one scenario to another, in such a way that solving the S concave problems by bundle methods is not expensive. This will be illustrated numerically in section 5.2.

After adding a row to G_k and an element to b_k , we increase the size of μ^k by adding one zero variable. Note that this does not impact the best bounds, since we have $\Theta_{k+1}((\mu^k, 0), \nu_1^k, \nu_2^k) = \Theta_k(\mu^k, \nu_1^k, \nu_2^k)$. In practice, we also expand the subgradients stored in the current memory of the bundle method used to maximize Θ_k . We can then warmstart the maximization of Θ_{k+1} with this new bundle information.

Stopping criteria. When hitting step 5, the algorithm stops looping over k considering the current model \check{v}_k as sufficiently rich. The test of step 5 is then a pragmatic rule to stop the algorithm, testing that the model of the recourse function cannot be improved much around the current iterate. Obviously, expecting x^k to be the optimal solution of (10) is excessive, in view of the nonconvexities of the problem. The stopping test still has an intrinsic meaning for a convexified version of our problem. This is studied in section 4.

Optional post-treatment The heuristics of step 2 are quick procedures that might not be entirely satisfactory. We describe them in appendix B. At the end of the algorithm, we might want to employ a more lengthy procedure to improve the solution. For instance we could employ an augmented Lagrangian based heuristic (e.g., [2, 56]). Notice that this aims at decreasing Δ_G^k but at the risk of increasing Δ_A^k . Therefore we quantify the changes to decide to accept the newly generated solution or retain the one that triggered the stopping test. To do so, we run one last additional model enrichment step and compare the sum of gaps $\Delta_G + \Delta_A$.

3.4 Illustration on toy-problem

We illustrate here the behaviour of our algorithm on a toy example. A complete numerical study on large-scale unit-commitment instances will be presented in section 5. The toy problem will also be used as an example in section 4.3.

Description of the problem. The toy generation park has two production units ($i = 1, 2$) and two periods ($t = 1, 2$ and $\tau = 1$). The first unit has a production $y_1 \in \{0, 3\}$ with cost $5y_1$ and the second $y_2 \in [0, 1]$ with cost $10y_2$. For each period, we want a total production matching a load $D = 2$ with bounds $s^d = -1$ and $s^u = 1$. Moreover, the first unit has the constraint that the production is constant over the two periods. Thus we have for this example

$$\begin{aligned} X^1 &= \left\{ (0, x_2^1, 0, x_2^2), (3, x_2^1, 3, x_2^2) : (x_2^1, x_2^2) \in [0, 1]^2 \right\} \\ X^2 &= \left\{ (x_1^1, x_1^2, x_2^1, x_2^2) : -1 \leq D - x_1^t - x_2^t \leq 1 \text{ for } t = 1, 2 \right\}. \end{aligned}$$

Observe that there are only two feasible solutions as $X^1 \cap X^2 = \{(0, 1, 0, 1), (3, 0, 3, 0)\}$. For the second stage, we generate a 100 load scenarios for $D(\xi)$ uniformly in $[1, 2]$. Since it does not satisfy the constraint $1 \leq D(\xi) - y_1^t + y_2^t$, the solution $(3, 0, 3, 0)$ not feasible for the second-stage problems. Therefore the optimal solution of this simple instance of (5) is $(0, 1, 0, 1)$.

Run of the algorithm. We run the algorithm on this example for illustrating its behaviour. Convergence to the optimal solution is obtained in three iterations. At the first iteration, the first stage generates the non-feasible schedule $(1, 0, 1, 0)$ and the primal recovery heuristic finds $(3, 0, 3, 0)$, which is indeed a feasible solution as seen from the first stage. The second stage detects infeasibility of this solution and a suboptimal cut is added. During the second iteration, the bundle method generates $(0, 1, 0, 1)$, which is also the retained primal iterate. The corresponding cut is added to the model. The third (and last) first stage iteration produces $(0, 1, 0, 1)$ as candidate solution, which automatically triggers to the stopping criteria.

4 Putting the method into perspective

In this section, we analyze the theoretical properties of the algorithm: its convexification effect (in section 4.1) and its convergence (in section 4.2). We use standard techniques of convex analysis, and refer frequently to [28], but we will end up with subtle properties. In particular, we will see in section 4.3 that our dual approach does not convexify at best the recourse function when the objective function is not linear.

4.1 Convexification effect of the algorithm

We study here the role of a convexified recourse function which sheds light on the behaviour of the algorithm. Our analysis features the convex envelope of the sum of f and the indicator function of X^1 , denoted $f_{X^1}^{**}$.

Such “restricted biconjugates” are a standard tool, tracing back to [19], studied intensively in [32], and already used in [17] in the context of stochastic optimization.

We introduce the convexified recourse cost function $\bar{c} : \mathbb{R}^n \times \mathbb{R}^k \rightarrow \mathbb{R} \cup \{+\infty\}$ as the optimal value of the following (convex) problem with $f_{X^1}^{**}$ as the objective function and $\text{conv } X^1$ replacing X^1 in the constraints

$$\bar{c}(x, \xi) := \begin{cases} \inf_{y \in \mathbb{R}^n} & f_{X^1}^{**}(y) \\ \text{s.t.} & y \in \text{conv}(X^1) \cap X^2(\xi) \\ & Px = Py. \end{cases} \quad (13)$$

In the general convex case, the role of $\bar{c}(\cdot, \xi)$ is key in our approach, so we formalize in the next proposition its main properties.

Proposition 1 (Convex recourse cost) *For a couple $(x, \xi) \in \mathbb{R}^n \times \mathbb{R}^k$, $\bar{c}(x, \xi)$ is the optimal value of the dual problem*

$$\bar{c}(x, \xi) = \sup_{(\lambda_1, \lambda_2, \lambda_3) \in \Lambda} \theta_{x, \xi}(\lambda_1, \lambda_2, \lambda_3) \in \mathbb{R} \cup \{+\infty\}. \quad (14)$$

Moreover, for any dual variables $(\lambda_1, \lambda_2, \lambda_3) \in \Lambda$, we have

$$c(x, \xi) \geq \bar{c}(x, \xi) \geq \lambda_1^\top Px + \lambda_2^\top (s^d - D(\xi)) + \lambda_3^\top (D(\xi) - s^u) + \bar{\theta}(\lambda_1, \lambda_2, \lambda_3). \quad (15)$$

The function $\bar{c}(\cdot, \xi)$ is closed and convex with respect to x , and if there exists a $(\lambda_1(x, \xi), \lambda_2(x, \xi), \lambda_3(x, \xi))$ attaining the sup in (14), we have $P^\top \lambda_1(x, \xi) \in \partial \bar{c}(x, \xi)$.

Proof By definition of \bar{c} , it follows from [32, Theorem 2.11/2.12] that $\theta_{x, \xi}$ defined in (7) is also the dual function of the problem (13). Now the compactness of X^1 allows us to apply a inf/sup theorem as [42, Cor.37.3.2] to the Lagrangian function associated with (13) to show that there is no duality gap, i.e., the equality (14). The equality also implies (15), by noting that $\bar{c}(x, \xi) \leq c(x, \xi)$ (as it can be seen from their respective definitions (3) and (13), and the fact that $f_{X^1}^{**} \leq f + i_{X^1}$). Expressed as a sup in (14), the convexity and closedness of $\bar{c}(\cdot, \xi)$ with respect to x is clear from [28, IV.2.1.2]. To get the value of the subgradient at a fixed (x, ξ) , we develop (15) taken with z and $(\lambda_1(x, \xi), \lambda_2(x, \xi), \lambda_3(x, \xi))$

$$\begin{aligned} \bar{c}(z, \xi) &\geq \lambda_1(x, \xi)^\top P(z - x) + \lambda_1(x, \xi)^\top Px + (\lambda_3(x, \xi) - \lambda_2(x, \xi))^\top D(\xi) + \bar{\theta}(\lambda_1(x, \xi), \lambda_2(x, \xi), \lambda_3(x, \xi)) \\ &\geq \lambda_1(x, \xi)^\top P(z - x) + \bar{c}(x, \xi), \end{aligned}$$

which ends the proof. \square

This result has two important consequences for our developments. As a first consequence, we directly get that the convexified expected recourse function defined by

$$\bar{v} : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}, \quad \bar{v}(x) := \mathbb{E}(\bar{c}(x, \xi)) \quad (\leq v(x)).$$

is convex and that we have a subgradient $P^\top \mathbb{E}(\lambda_1(x, \xi)) \in \partial \bar{v}(x)$ at $x \in \text{dom}(\bar{v})$. This yields that the cutting planes model \check{v}_k of (8)-(9) is not only an inexact cutting plane model for v : it is more precisely an (exact) cutting planes model for \bar{v} as

$$\check{v}_k(x) = \max_{i=1, \dots, k-1} \left\{ (g^i)^\top (x - x^i) + \bar{v}^i \right\} \leq \bar{v}(x) \leq v(x), \quad (16)$$

and moreover when $x^i \in \text{dom}(\bar{v})$, then $\bar{v}^i = \bar{v}(x^i)$. Thus, in step 3 of the algorithm, a linearization of \bar{v} is computed, and our algorithm sees in fact only \bar{v} (and not v). This means that our decomposition method, though using only objects defined from data of our initial problem (5), solves implicitly

$$\begin{aligned} \min_{x \in \mathbb{R}^n} & f(x) + \bar{v}(x) \\ \text{s.t.} & x \in X^1 \cap X^2. \end{aligned}$$

This will be detailed further in section 4.2. It is nevertheless important to note that \bar{v} is just a convex surrogate of v , but not its convex hull, as shown in section 4.3.

Before moving to these two points, we emphasize the second important consequence of Proposition 1, about the implicit constraint in (17). When the iterate x^i does not belong to the domain of \bar{v} , i.e., there exists a scenario ξ_ℓ such that $x^i \notin \text{dom}(c(\cdot, \xi_\ell))$, then Proposition 1 gives that

$$\sup_{(\lambda_1, \lambda_2, \lambda_3) \in A} \theta_{x^i, \xi_\ell}(\lambda_1, \lambda_2, \lambda_3) = +\infty. \quad (17)$$

Let us now argument that the fact that θ_{x^i, ξ_ℓ} tends to $+\infty$ when x^i does not belong to the domain of \bar{v} implies that the (sub)optimality cuts in (16) act like feasibility cuts for $\text{dom}(\bar{v})$.

Proposition 2 ((Sub)optimality cuts act like feasibility cuts) *Assume that we know a bound $\Delta > 0$ on the duality gap*

$$\Delta_G^k \leq \Delta \quad \text{for all } k \geq 1,$$

and a bound $M > 0$ on the objective function

$$\max_{x \in X^1 \cap X^2} f(x) + v(x) \leq M.$$

If $x^i \notin \text{dom}(\bar{v})$, then the (sub)optimality cut (9) allows the algorithm to cut off the point x^i (i.e., $x^k \neq x^i$ for all $k \geq i$), provided that θ_{feas} is large enough, more precisely

$$\theta_{\text{feas}} \geq (M + \Delta) / \min\{p_1, \dots, p_S\}. \quad (18)$$

Proof Let $x^i \in X^1 \cap X^2$ and $\xi_\ell \in \Xi$ be such that $x^i \notin \text{dom}(c(\cdot, \xi_\ell))$. We have (17) and then the bundle algorithm maximizing $\theta_{x, \xi}$ in (6) will stop with the test $\theta_{x^i, \xi_\ell}(\lambda_1^i, \lambda_2^i, \lambda_3^i) \geq \theta_{\text{feas}}$. Let now $k > i$; we write

$$\begin{aligned} f(x^i) + \check{v}_k(x^i) &\geq \bar{v}^i && \text{[by definition of } \check{v}_k \text{ and since } f \geq 0] \\ &= \sum_j p_j \theta_{x^i, \xi_j}(\lambda_1^i, \lambda_2^i, \lambda_3^i) && \text{[by (9), the definition of } \bar{v}^i] \\ &\geq p_\ell \theta_{x^i, \xi_\ell}(\lambda_1^i, \lambda_2^i, \lambda_3^i) && \text{[since } \theta_{x^i, \xi_j} \geq 0] \\ &\geq \min\{p_1, \dots, p_S\} \theta_{x^i, \xi_\ell}(\lambda_1^i, \lambda_2^i, \lambda_3^i) && \text{[by definition of } p_{\min}] \\ &> M + \Delta. && \text{[by (18)]} \end{aligned}$$

By definition of M , this shows that x^i cannot be an Δ -solution of the approximate problem (10) at iteration k . This implies that x^k cannot be x^i , otherwise it would contradict (12). \square

In practice, we have a reasonable idea of M and Δ . Note finally that they could be defined dynamically: in such a situation one would define

$$\theta_{\text{feas}} \geq (f(\hat{x}^i) + \check{v}(\hat{x}^i) + \Delta) / p_{\min},$$

where \hat{x}^i is akin to the current best feasible point. As a consequence of such a setting, points that lack interest (such that $\theta_{x^i, \xi_\ell}(\lambda_1^i, \lambda_2^i, \lambda_3^i) \geq \theta_{\text{feas}}$), can no longer be distinguished clearly from points $x^i \notin \text{dom}(\bar{v})$. This updating rule is similar to the one of the on-demand accuracy bundle of [15].

4.2 Convergence analysis

This section provides a convergence analysis of the algorithm. As previously explained, the algorithm uses dual approaches for both first and second stages that cannot distinguish between v and its convex counterpart \bar{v} . This allow us to establish guarantees only for the convexified problem (17). The first result completes (12) about the quality of x^k at each iteration.

Proposition 3 (Approximate solution) *If $x^k \in \text{dom}(\bar{v})$, then the approximation error defined in the algorithm satisfies*

$$\Delta_A^k = \bar{v}(x^k) - \check{v}_k(x^k). \quad (19)$$

Consequently, x^k is a $(\Delta_A^k + \Delta_G^k)$ -solution of the convexified problem (17).

Proof Observe that the expression of \check{v}_k in (8) gives that $\check{v}_{k+1}(x^k) = \max\{\check{v}_k(x^k), \bar{v}^k\}$, which in turn gives

$$\Delta_A^k = \check{v}_{k+1}(x^k) - \check{v}_k(x^k) = \max\{0, \bar{v}^k - \check{v}_k(x^k)\}.$$

When $x^k \in \text{dom}(\bar{v})$, we have $\bar{v}^k = \bar{v}(x^k) \geq \check{v}_k(x^k)$, so that (19) holds. Finally, we deduce from (12) and (19) that for $x \in X^1 \cap X^2$,

$$f(x) + \bar{v}(x) \geq f(x) + \check{v}_k(x) \geq f(x^k) + \check{v}_k(x^k) - \Delta_G^k \geq f(x^k) + \bar{v}(x^k) - \Delta_G^k - \Delta_A^k.$$

The above inequality means that x^k is optimal up to $\Delta_G^k + \Delta_A^k$ for problem (17). \square

This gives a better understanding of our stopping test and the following optional improvement step. Roughly speaking, the stopping criteria means that the cutting-plane model is nearly optimal. However it does not give a controllable approximate solution because of the gap error Δ_G^k in the above lemma. The gap error is obtained by running heuristics, and, though these heuristics perform well in practice, they have no theoretical guarantee. The optional post-treatment step aims at improving the quality of the solution by decreasing Δ_G^k .

The second result of this section establishes the convergence of the algorithm, under a technical assumption. The analysis follows usual rationale of standard cutting-plane methods in a convex setting (see e.g., [29, XII.4.2] and [43, Theorem 7]).

Theorem 1 (Asymptotic convergence) *Assume that the algorithm generates a sequence, which after finitely many iterations, is all contained in a compact set C lying in the relative interior of $\text{dom}(\bar{v})$. Then, the algorithm terminates after finitely many iterations, and the final iterate x_{final} is a $\delta_{\text{tol}} + \Delta_G^{\text{final}}$ -optimal solution for (17).*

Proof For all $x \in C$, there exist a $\varepsilon_x > 0$ such that the ball of center x and radius ε_x is included in the relative interior of $\text{dom}(\bar{v})$. Extracting a finite number of balls recovering the compact set C , we can strengthen the assumption as: there exists a $\varepsilon > 0$ such that the compact set

$$C_\varepsilon = \{x + b : x \in C, \|b\| \leq \varepsilon\}$$

lies in the relative interior of $\text{dom}(\bar{v})$. Together with the convexity of \bar{v} (consequence of Proposition 1), this gives us that \bar{v} is Lipschitz-continuous on C_ε ([29, IV.3.1.2]); let us denote by $L > 0$ the Lipschitz constant.

By assumption, there exists an index $K \geq 0$ such that $x^k \in C \subseteq \text{int dom}(\bar{v})$ for all $k \geq K$. For sake of a contradiction, assume that the method does not terminate, which means that $\Delta_A^k > \delta_{\text{tol}}$ for all $k > K$. Since $x^k \in \text{dom}(\bar{v})$, (19) yields

$$\delta_{\text{tol}} < \bar{v}(x^k) - \check{v}_k(x^k).$$

Let us use the model \check{v}_k explicitly, by noting that for all $K \leq i < k$

$$\bar{v}(x^i) + (\bar{g}^i)^\top (x^k - x^i) \leq \check{v}_k(x^k)$$

which yields, by the Cauchy-Schwarz inequality,

$$\delta_{\text{tol}} < \bar{v}(x^k) - \bar{v}(x^i) + \|\bar{g}^i\| \|x^k - x^i\|. \quad (20)$$

Now note that the subgradient inequality gives in particular that

$$\bar{v}(x^i + \varepsilon \bar{g}^i / \|\bar{g}^i\|) \geq \bar{v}(x^i) + \varepsilon \|\bar{g}^i\|,$$

which, by Lipschitz-continuity on the set C_ε , implies that $\|\bar{g}^i\| \leq L$. Finally, using the Lipschitz-continuity of \bar{v} again, we get from (20)

$$\|x^k - x^i\| > \delta_{\text{tol}} / 2L$$

which contradicts the fact that we can extract a converging subsequence from (x^k) belonging to the compact set C . Thus, the algorithm terminates, and the final approximation of comes from Proposition 3. \square

Arguably the convergence result is limited since it relies on an ultimately relative recourse like assumption. Looking closely to the proof, we see that the key implication of this assumption is that there exists a bound on the subgradients: $\|\bar{g}_i\| \leq L$ for all i . Such an assumption is standard in the analysis of cutting-plane method in convex case. For example, it appears explicitly in [43, Assumption 6]. It also holds in the analysis of generalized Benders decomposition [26, Theorems 2.4, 2.5, Lemma 2.1].

4.3 On the convex envelope of the recourse function

As explained in the previous sections, the algorithm has a convexification effect featuring \bar{v} , which is a “convex surrogate” of v . We show here how \bar{v} relates to the convex envelope of v . Fix $\xi \in \Xi$ and introduce $\tilde{c} : \mathbb{R}^n \times \mathbb{R}^k \rightarrow \mathbb{R} \cup \{+\infty\}$ defined by

$$\tilde{c}(x, \xi) := \begin{cases} \min_{y \in \mathbb{R}^n} & f_{X^1}^{**}(y) \\ \text{s.t.} & y \in \text{conv}(X^1 \cap X^2(\xi)) , \\ & Px = Py, \end{cases}$$

as well as its expectation $\tilde{v} : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{+\infty\}$ defined by $\tilde{v}(x) := \mathbb{E}(\tilde{c}(x, \xi))$. These functions also provide convex surrogates of the recourse functions, as established by the next proposition.

Proposition 4 (Convexified functions) *For $x \in \mathbb{R}^n$ and $\xi \in \Xi$, we have $c(x, \xi) \geq \tilde{c}(x, \xi) \geq \bar{c}(x, \xi)$, and*

$$v(x) \geq \tilde{v}(x) \geq \bar{v}(x).$$

Moreover, $\tilde{c}(\cdot, \xi)$ and \tilde{v} are closed and convex. When X^1 is convex, we have in fact $\tilde{c}(\cdot, \xi) = \bar{c}(\cdot, \xi)$ and $\tilde{v} = \bar{v}$.

Proof The two inequalities come directly by the inclusions of the sets

$$X^1 \cap X^2(\xi) \subset \text{conv}(X^1 \cap X^2(\xi)) \subset \text{conv}(X^1) \cap X^2(\xi) \quad (21)$$

and the fact that $f_{X^1}^{**} \leq f$ on X^1 . Let us argue that $\tilde{c}(\cdot, \xi)$ is a closed convex function, then the result for \tilde{v} comes by integration. Write \tilde{c} as a “lower-bound function” (see e.g [28, IV]):

$$\tilde{c}(x, \xi) = \min\{r : (x, r) \in C\}$$

with $C = \{(x, r) \in \mathbb{R}^{n+1} : \exists y \in \text{conv}(X^1 \cap X^2(\xi)) \text{ such that } Px = Py \text{ and } f_{X^1}^{**}(y) \leq r\}$. Together with the properties of $X^2(\xi)$ and $f_{X^1}^{**}$, compactness of X^1 (and consequently of $\text{conv}(X^1 \cap X^2(\xi))$) yields that C is a closed convex set. As the lower-bound function of a closed convex set, \tilde{c} is a closed convex function (by [28, IV.1.3.1]). Finally, in the case when X^1 is convex, the inclusions of (21) are actually equalities, and then the functions are equal by definition. \square

The above lemma implies that the (closed) convex envelope of v is greater or equal to \tilde{v} . In fact equality holds in the linear case.

Proposition 5 *Assume that f is linear, then the map \tilde{v} is the closed convex envelope of v .*

Proof We just have to show that, for an fixed $\xi \in \Xi$ the map $x \mapsto \tilde{c}(x, \xi)$ is the closed convex envelope of $x \mapsto c(x, \xi)$. For convenience of notation, we drop the dependency on ξ in the proof. We start with noticing that c can be rewritten with the help of [17, Lem. 1] as

$$\tilde{c}(x, \xi) = \begin{cases} \min_{y \in \mathbb{R}^n} & f(y) \\ \text{s.t.} & y \in \text{conv}(X^1 \cap X^2(\xi)) , \\ & Px = Py, \end{cases}$$

because f is linear and $X^1 \cap X^2(\xi)$ compact. Let us show now that \tilde{c} is equal to the biconjugate $c^{**} = \text{conv } c$ (see [29, X.1.3.6]); the proof consists in establishing the equality between the convex conjugates of the two functions $-c^* = -(\tilde{c})^*$. We start with computing $-c^*$: for $\lambda \in \mathbb{R}^n$

$$-c^*(\lambda) = \min_{x \in \mathbb{R}^n} c(x) - \lambda^\top x = \min_{x \in \mathbb{R}^n} \min_{y \in X^1 \cap X^2, Px=Py} f(y) - \lambda^\top x = \min_{y \in X^1 \cap X^2} (f(y) - \sigma_{\{x: Px=Py\}}(\lambda))$$

where $\sigma_{\{x: Px=Py\}}$ is the support function of the affine space (in x) defined by the equation $Px = Py$. Thus we get that $-c^*(\lambda)$ is $-\infty$ unless λ lies in the image of P^\top , and that, in this case,

$$-c^*(P^\top \mu) = \min_{y \in X^1 \cap X^2} f(y) - \mu^\top Py.$$

Since f is affine, the above minimum can be taken on the convex hull of the constraints:

$$-c^*(P^\top \mu) = \min_{y \in \text{conv}(X^1 \cap X^2)} f(y) - \mu^\top P y. \quad (22)$$

Observe that the right-hand-side is also $-(\tilde{c})^*(P^\top \mu)$, since its expression can be derived in the very same way as we get (22). Thus we have $-c^* = -(\tilde{c})^*$, and conjugating a second time gives $c^{**} = (\tilde{c})^{**} = \tilde{c}$ using the fact that \tilde{c} is closed and convex (by Proposition 4). \square

In the general case though, the two functions \tilde{v} and \bar{v} are different: they do not have the same domain and also differ on the intersection of their domains. To see this, let us come back to the toy example of section 3.4 with only one load scenario $D(\xi) = 2$

$$v(x) = \begin{cases} \min_{(y_1^1, y_1^2, y_2^1, y_2^2) \in \mathbb{R}^4} & 5(y_1^1 + y_1^2) + 10(y_2^1 + y_2^2) \\ \text{s.t.} & (y_1^1, y_1^2) \in \{0, 3\}^2, (y_2^1, y_2^2) \in [0, 1]^2 \\ & -1 \leq 2 - y_1^t - y_2^t \leq 1 \text{ for } t = 1, 2 \\ & y_1^1 = y_1^2, y_1^1 = x_1, y_1^2 = x_2. \end{cases} \quad (23)$$

Explicit expressions of v , \tilde{v} and \bar{v} can be worked out by elementary considerations (skipped here): we have

$$v(x) = \begin{cases} 20 & \text{if } x = (0, 1) \\ 30 & \text{if } x = (3, 0) \\ +\infty & \text{elsewhere} \end{cases}$$

and its convexified counterparts are

$$\tilde{v}(x) = \begin{cases} 20 + 10x_1/3 & \text{if } x = (x_1, 1 - x_1/3) \\ +\infty & \text{elsewhere} \end{cases}$$

and, for $x \in X = \{x \in [0, 3] \times [0, 1] : 1 \leq x_1 + x_2 \leq 3\}$,

$$\bar{v}(x) = \max\{10x_2 + 10, 10(x_2 + x_1)\} = \begin{cases} 10x_2 + 10 & \text{if } x \in X \text{ and } x_1 \leq 1 \\ 10(x_2 + x_1) & \text{if } x \in X \text{ and } x_1 > 1 \\ +\infty & \text{elsewhere} \end{cases}$$

As illustrated in figure 1, we see that \tilde{v} is strictly above \bar{v} . In particular, when restricted to the open segment $\{(x_1, 1 - x_1/3), x_1 \in (0, 3)\}$, we have $\tilde{v}(x) = 20 + 10x_1/3 > \bar{v}(x) = \max\{20 - 10x_1/3, 10 + 20x_1/3\}$.

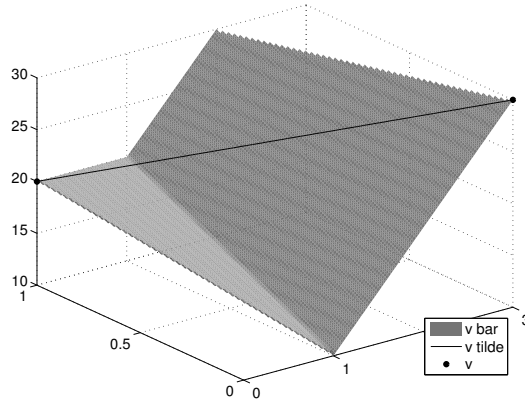


Fig. 1 Convexification of v for the toy example

5 Numerical Experiments

5.1 Experimental setting

Data set description. We consider a data set coming from the French unit-commitment problems of EDF's generation assets with 136 thermal units and 22 hydro valleys ($m = 158$) on a time horizon of two days discretized by half-hour ($T = 96$). Subproblems are modeled following classical operational formulations (see e.g., [47]): for thermal units, we set up a mixed-integer linear problem similar to the one of [25], and for hydro units, we have a linear problem as in [34, 52]. Details on the formulations of subproblems used in the numerical experiments are given in appendix. Overall, we end up with a deterministic problem with 47450 continuous variables, 26112 binary variables and 812906 constraints.

Load uncertainty. We generate the uncertain loads as an average load on top of which we add a causal time series model (see, e.g., [9]). More precisely, we consider the Gaussian random variable $D(\xi) = \bar{D} + \zeta$, where \bar{D} is an observed load and ζ is an $AR(3)$ model with coefficients $\varphi := (0.9, 0.6, -0.51)$ and Gaussian innovations. Writing the $AR(3)$ process in its causal time series representation (see e.g., [46]), the covariance matrix of $D(\xi)$ is $\Sigma^D = (C^D S)(S(C^D))^T$, where S is a diagonal matrix with element

$$S_{ii} = \mathbf{f} \frac{\bar{D}_i}{\frac{1}{T} \sum_{j=1}^T \bar{D}_j} \quad (24)$$

and C^D is the nominal covariance matrix with elements $C_{ji} = \sum_{k=1}^{j-i+1} \varphi_k^D$ defined from

$$\varphi_k^D = \begin{cases} \varphi_3 & \text{if } k = 1, \\ \varphi_3 \varphi_1^D + \varphi_2 & \text{if } k = 2, \\ \varphi_3 \varphi_2^D + \varphi_2 \varphi_1^D + \varphi_1 & \text{if } k = 3, \\ \sum_{k=1}^3 \varphi_{4-k} \varphi_{j-k}^D & \text{otherwise.} \end{cases}$$

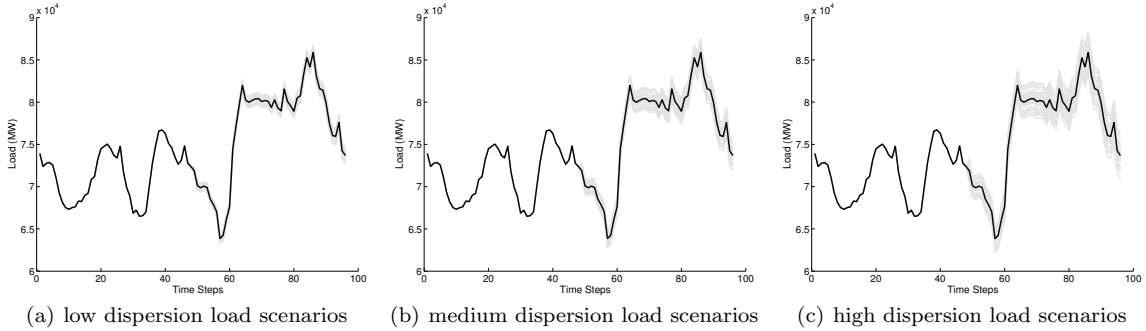


Fig. 2 Some generated load scenarios with three level of dispersion

Stochastic unit-commitment instances. In (24), the parameter \mathbf{f} is a factor reflecting the load dispersion. By setting this parameter to three different values, we create three stochastic unit-commitment: a low-dispersion instance (with $\mathbf{f} = 1.1$), a medium-dispersion instance (with $\mathbf{f} = 2$), and a high-dispersion instance (with $\mathbf{f} = 3$). Figure 2 plots several load scenarios for each level of dispersion. With 50 generated scenarios, each data set gives a global optimization problem (5) with 1209975 continuous variables, 665856 binary variables and 20729103 constraints, which is out of reach for mixed-integer linear solvers.

Algorithm. On the three problems, we run the algorithm of section 3.3, with the following setting. We set the stopping threshold to $\delta_{\text{tol}} = 1\%$. We also set to 300 the size of the limited memory of the second stage bundle (more precisely, this is the size of the storage of the bundle information on $\bar{\theta}$ of (7)). We observed that this choice offers a good trade-off for the bundle algorithm between the number of iterations and the cost of its quadratic subprograms. Finally, for the step 3 of the algorithm, we use the four following Lagrangian heuristics (described in appendix):

- CTI: the commitment based heuristic [7] with time independent priority list,
- CTD: the same as above but with a time dependent list,
- RH: the recombining heuristic of [48],
- allH: taking the best of the three above heuristics.

5.2 Numerical results

Results with 50 scenarios. Table 1 presents the computational results on the three stochastic unit-commitment described in the previous section (with $S = 50$ scenarios and with respectively low, medium, and high dispersion of the stochastic load). Let us point out three features in these figures. First we observe that the final gaps are rather small, often lower than 1%, and quite comparable to those observed when solving deterministic unit-commitment problems. Second, the primal recovery heuristics give different results but without significant changes in the number of calls or in the final gaps. Notice that the result for the hybrid heuristic combining the three others gives the same results as the CTI for the instances with small and large dispersion. More precisely, the heuristic providing the best results is always CTI, except for a single iteration when processing the medium instance. Third, we emphasize that the number of oracle calls remains within reasonable limits. For comparison, using up to 300 oracles calls for solving deterministic unit-commitment problems is common (e.g. [18, Table 3.1]). In our case we have provided a good initial guess for the dual signals and the bundle method converged in 72 iterations.

Instance	Heuristic used in Step 3	Nb. Iter	Oracle Calls		Gaps (%)		$f(x^{\text{final}})$
			1st Stage	2nd Stage	Δ_G^{final}	Δ_A^{final}	
low	CTD	8	361	1741	1.48	0.66	5.645×10^7
	CTI	8	417	1496	0.27	0.54	5.541×10^7
	RH	6	267	1043	0.63	0.87	5.583×10^7
	allH	8	417	1496	0.27	0.54	5.541×10^7
medium	CTD	8	404	2441	1.59	0.73	5.648×10^7
	CTI	9	445	2238	0.31	0.91	5.559×10^7
	RH	11	458	2607	1.58	0.40	5.608×10^7
	allH	14	776	3501	0.60	0.72	5.552×10^7
high	CTD	11	516	3817	1.66	0.71	5.648×10^7
	CTI	6	293	1553	0.19	0.71	5.547×10^7
	RH	8	317	2374	0.67	0.62	5.584×10^7
	allH	6	293	1553	0.19	0.71	5.547×10^7

Table 1 Numerical results of the algorithm using the four different heuristics

Effect of hot-starting. Our algorithm solving two-stage unit-commitment problems shows similar results and similar order of numerical complexity as a bundle algorithm solving deterministic unit-commitment problems. Recall though that our algorithm solves $(S+1)$ full unit-commitment problems at each iteration. The main reason for this is the efficiency of the hot-starting procedure for the model enrichment step.

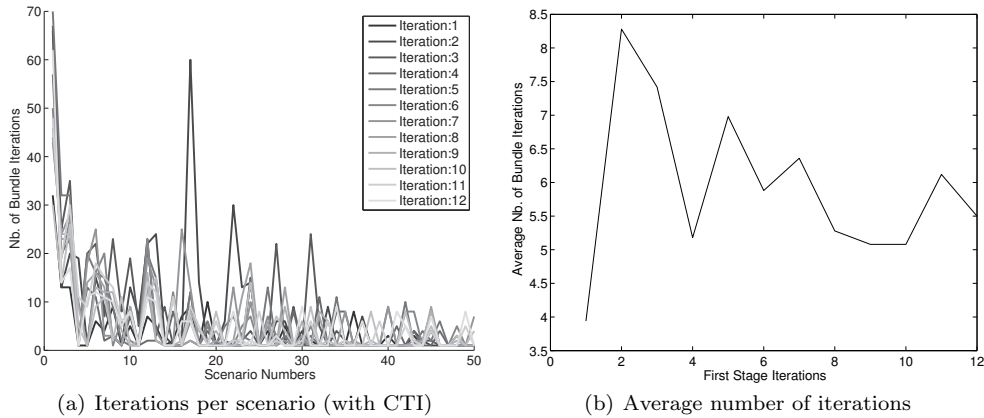


Fig. 3 Required Iterations for each model enrichment step

Let us illustrate further this efficiency by showing on Figures 3(a) and 3(b) the number of iterations of the second-stage for one run of the algorithm on the first instance. We observe that a change of iterates implies that early scenarios require several iterations, but that this effect quickly diminishes and only few additional iterations are needed for other scenarios. The overall computational effort remains rather constant as seen on Figure 3(b): around only 5 bundle iterations are needed in order to process a single scenario. The whole set of 50 scenarios can therefore be processed within approximately 250 bundle iterations. We recall that performing 10 iterations of the algorithm, while using 50 scenarios for representing uncertainty gives a global problem size equivalent to solving 510 full large scale unit-commitment problems.

Larger instances with more scenarios. We now provide some more results with an increased number of scenarios $S \in \{50, 100, 250\}$. Let us focus on the high-dispersion instance and the CTI heuristic. Table 2 shows that the average number of oracle calls per iteration and scenario decreases as the number of scenarios increases. It is remarkable that the total number of oracle calls stays reasonable. Notice that we helped the bundle method by increasing the size of the memory of the bundle from 300 to 500 for this experiment. This explains the difference between the first line of Table 2 with the tenth line of Table 1.

Number of Scenarios	Nb. iter	Oracle calls		Average oracle call
		1st Stage	2nd Stage	
50	4	167	1009	5.88
100	8	360	3461	4.77
250	16	694	14205	3.73

Table 2 Larger instances with more scenarios: decrease of oracle calls per scenario and iteration

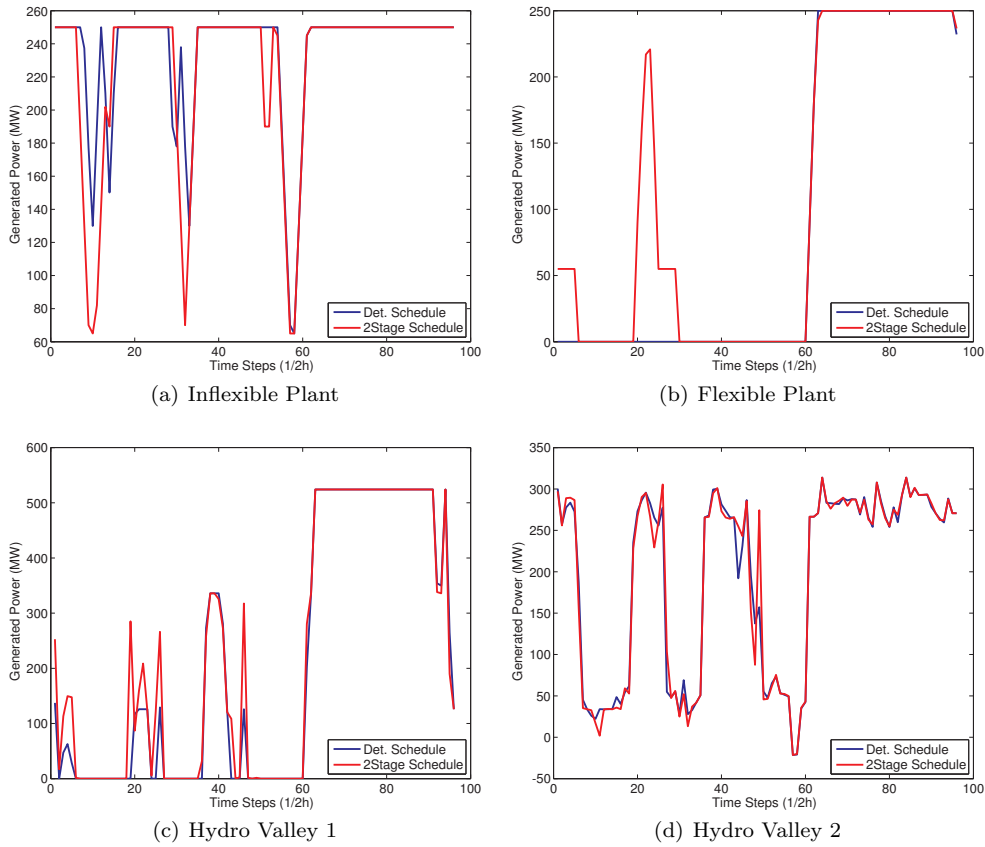


Fig. 4 Comparison of generation schedules given by our two-stage formulation and the deterministic one.

Effect on the generation planning. We illustrate the typical effect of our two-stage approach on the final generation plannings, compared to the ones computed by a deterministic model. We see on Figures 4(a), 4(b), and 4(c) below that generation is transferred from inflexible but cheap production sources to more expensive but flexible sources. This includes significant changes for several hydro valleys as shown in Figure 4(c). However most hydro valleys retain a similar shaped production profile as illustrated in Figure 4(d).

A variant of the algorithm with multi-cuts Recall that we use a cutting-plane model for v aggregating the cuts for each scenario in (9). Another option would be to set up a cutting-plane model for each scenario. In this case, the resulting model for v would be tighter, so the convergence possibly faster. On the other hand, the size of the dual multipliers μ for problem (10) would be multiplied by S scenarios, and therefore the first-stage dual problems are in larger dimension. In a final experiment (with $S = 50$ scenarios), we investigate the effect of using a multi-cut variant of our algorithm. We take a tolerance $\delta_{\text{tol}} = 0.25\%$, smaller than before, to have more iterations and then better distinguish the effect of multi-cuts. Table 3 presents the results. The column related to the iteration ratio shows that occasionally the number of iterations increases by 30% when using the multi-cut version, but the opposite phenomena appears as well. We thus observe that using a multi-cut approach is not always beneficial. The columns related to the first and second stage cost report the ratio of the total number of oracle calls per stage normalized by the total number of iterations. On average the second stage cost is roughly identical, whereas the first stage cost of the multi-cut version is around 12% less than the mono-cut version. Thus, the increased size of the dual multipliers does not seem to be an issue. Note that the seemingly very poor performance of the multi-cut version on the medium instance with the CTD heuristic is due to a very sudden jump in precision in the mono-cut version. This leads to early termination, whereas in the multi-cut version progression towards the stopping criteria is far more gradual. Note that table 3 seems to indicate that the increased precision of the multi-cut model is somehow offset by the primal recovery heuristics, even though they perform similarly well in both situations. The fact that disaggregating cutting plane methods does not always lead to a decrease of the number of iterations was also observed in deterministic unit-commitment, in e.g., [51].

Instance	Heuristic	Iteration increase (%)	1st Stage cost (%)	2nd Stage cost (%)
low	CTI	30.0	3.66	1.95
	CTD	33.3	-19.80	-5.11
	RH	25.0	2.97	-1.96
	allH	-52.5	-9.42	-2.42
medium	CTI	44.4	0.36	6.13
	CTD	330.0	4.44	-2.13
	RH	-38.5	0.82	1.45
	allH	44.4	0.36	6.13
high	CTI	52.6	-49.37	-12.61
	CTD	-22.2	-27.27	-6.14
	RH	12.5	-1.44	0.60
	allH	89.5	-50.61	-12.32
Average		45.72	-12.11	-2.20

Table 3 Numerical results of the algorithm vs its multi-cuts version : ratio of the number of iterations increase (iteration increase), the difference of oracle calls per stage normalized by the total number of iterations for the first stage (1st stage cost), and the same difference for the second stage (2nd stage cost).

6 Conclusion

In this paper, we have proposed a decomposition algorithm for solving two-stage problems, where both first and second stages are full unit-commitment problems. The algorithm makes no specific assumptions on the set of technical constraints for the units and only uses tools already well in place for deterministic unit-commitment. The working horses of the approach are several hot-started bundle methods with a small overall computational burden. We have analyzed the convergence of the algorithm and studied its convexifying effect. We have shown its efficiency on real life unit-commitment instances, for which a remarkably low number of iterations and oracle calls are needed to reach convergence.

There is room for improvement in the algorithm and its current implementation. It would be interesting to investigate using standard regularization techniques, in particular first stage bundle-like regularizations and second stage augmented Lagrangian regularizations. This would nevertheless add a layer of complexity on the presentation, the theoretical study, and the implementation of the algorithm. This is beyond the scope of this paper, and we defer it to future research. Other points of improvement of the implementation would include the preconditioning of the second-stage bundle [1] and the use of available uncontrolled bundle information [33].

Acknowledgment

We thank the three referees for their comments and suggestions, that helped us to simplify and clarify the presentation of our work. We acknowledge the support of the Gaspard Monge program for Optimization and Operations Research (PGMO) Project "Consistent Dual Signals and Optimal Primal Solutions". The second author also acknowledges the support of the CNRS Mastodons project "gargantua/titan" and of the Labex Persyval-Lab project-action "Khronos".

A Description of the unit-commitment model

This appendix provides more information about the models of subproblems used in the numerical experiments of section 5.

Hydro valley subproblems. The hydro valley subproblems deal with optimizing the power production of turbines and pumps for a given price signal. The turbines and pumps connect various reservoirs together. For a given topology, one readily establishes the flow equations that deal with updating the reservoirs levels through time. These reservoir levels have to remain between a lower and upper bound for all T time steps. Turbines and pumps are moreover subject to natural bounds on production levels. The most challenging feature to take into account is the turbinning efficiency function that associates with each turbin quantity (m^3/h) and water head (reservoir level in uphill reservoir, in m^3) the amount of produced power (MW). This function can be highly non-linear, non-concave and may even contain forbidden zones of production, see e.g. [8, 22].

A common assumption in the French system (see, e.g., [34]) is that the water-head effect for large reservoirs can be neglected as the volumetric difference that can be achieved during the T time steps is quite small. For smaller reservoirs the effect caused on the amount of produced power would be quite small. Moreover following the set of assumptions made in [34], the power efficiency function becomes concave and is approximated with an a priori piecewise linearization. This makes the hydro valley subproblem a linear program. More details can be found in [52].

Thermal subproblems. As the thermal subproblems are concerned, we set up a usual model, similar to the one of [25], that incorporates, minimum generation levels, minimal up and down times, start up costs, fixed generation costs and ramping rates. We provide here a short description for convenience. To simplify notation, we do not include a reference to the specific unit the problem belongs to,

The decision variables are $p \in \mathbb{R}_+^T$ providing the amount of generated power in MW , $u \in \{0, 1\}^T$ the on/off status of the unit for each time step and $z \in \{0, 1\}^T$ an auxiliary variable indicating an effective start of the unit. Problem data describing cost are $c \in \mathbb{R}_+^T$ in $\text{€}/MW$, a proportional cost of production, $c_f \in \mathbb{R}_+^T$ in $\text{€}/h$, a fixed production cost and $c_s \in \mathbb{R}_+^T$ in € , a start up cost. Bounds on production levels expressed in MW , when producing are given by $p_{\min} \in \mathbb{R}_+^T$ and $p_{\max} \in \mathbb{R}_+^T$. Ramping rate related data is $g_+, g_- > 0$ expressed in MW/h and correspond to the ramping up gradient and ramping down gradient respectively. The numbers $s_+, s_- > 0$ express similar quantities but for starting and stopping ramping rates. Finally τ_+, τ_- expressed in a number of time steps correspond to the minimum up and down times respectively. We make the assumption that when a unit is online for exactly τ_+ time steps the minimum up constraint is satisfied (while [25] assume this for $\tau_+ + 1$ time steps). The optimization problem can then be stated as follows, where $\lambda \in \mathbb{R}^T$ is a Lagrangian price multiplier ($\text{€}/MW$):

$$\begin{aligned} \min_{p, u, z \in \mathbb{R}_+^T \times \{0, 1\}^{2T}} \quad & (c - \lambda)^T p \Delta t + c_f^T u \Delta t + c_s^T z \\ \text{s.t.} \quad & p_{\min}(t)u(t) \leq p(t) \leq p_{\max}(t)u(t), \quad \forall t = 1, \dots, T \\ & p(t) \leq p(t-1) + u(t-1)g_+ \Delta t + (1 - u(t-1))s_+ \Delta t, \quad \forall t = 1, \dots, T \\ & p(t-1) \leq p(t) + u(t)g_- \Delta t + (1 - u(t))s_- \Delta t, \quad \forall t = 1, \dots, T \\ & u(t) \geq u(r) - u(r-1), \quad \forall t = t_0 + 1, \dots, T, r = t - \tau_+ + 1, \dots, t - 1 \\ & u(t) \leq 1 - u(r-1) + u(r), \quad \forall t = t_0 + 1, \dots, T, r = t - \tau_- + 1, \dots, t - 1 \\ & u(t) - u(t-1) \leq z(t), \quad \forall t = 1, \dots, T. \end{aligned}$$

Here Δt corresponds to the size of each time step expressed in hours, $p(0)$ to the initial power output and t_0 is defined according the amount of time τ_0 (in time steps) the unit has spend producing or is offline. More specifically,

$$t_0 = \begin{cases} \max\{0, \tau_+ - \tau_0\} & \text{if } p(0) > 0 \\ \max\{0, \tau_- - \tau_0\} & \text{otherwise} \end{cases}$$

Obviously $u(0) = 1$ in the first case and $u(0) = 0$ in the second.

B Lagrangian Heuristics

As in decomposition approaches for deterministic unit-commitment, heuristics play an important role in our algorithm – more precisely in Step 2. In our numerical experiments, we use three heuristics inspired from [7] and [48]. This section describes them briefly.

Three Lagrangian heuristics. The three heuristic use information returned by the bundle algorithm maximizing (11) used in Step 1. More precisely, denoting by p is the number of iterations of this algorithm and x^j a primal iterate obtained at iteration $j \in \{1, \dots, p\}$, the heuristics use the following quantities:

1. the dual simplicial multipliers α of the quadratic program solved at the last iteration of the bundle method;
2. the so-called pseudo schedule \hat{x} is defined as $\sum_{j=1}^p \alpha_j x^j$, see [14, 18];
3. the pseudo costs $(\hat{c}_1, \dots, \hat{c}_m)$ defined as $\hat{c}_i = \sum_{j=1}^p \alpha_j c_i^j$ where c_i^j the pure production cost of subproblem i at iteration j ;
4. the pseudo commitment decisions defined as $\hat{u}_i^j = \sum_{j=1}^p \alpha_j u_i^j$, where $u_i^j \in \{0, 1\}^T$ are the commitment decisions of each thermal plant for each iteration $j = 1, \dots, p$.

Another common ingredient is the resolution of an economic dispatch problem: for a fixed set of commitment decisions, we let a continuous optimization problem adjust production levels in order to generate a solution in X^2 .

We begin by remarking that the pseudo-schedule is a technically feasible solution as hydro valleys are concerned (since these sub-problems have convex feasible sets, see the previous section). Also the pseudo-schedule is directly related to offer-demand equilibrium constraints through bundle stopping criteria. We therefore keep the pseudo-schedule as hydro-valleys are concerned and remove their generation from the load D in order to form \tilde{D} . The production parc is such that the obtained net load is always strictly positive. The heuristics are therefore mostly concerned with thermal plants. For convenience of notation we will still use m to index the number of thermal plants.

Commitment based heuristic. This heuristic is inspired from [7]. We begin with an initial guess for the commitment decisions called \tilde{u} , for instance one of the commitment decisions encountered during the optimization of problem (11). We now build a priority list in two different ways. The first is time-independent and related to sorting the pseudo costs divided by total generated pseudo power in increasing order. A lower value indicates a unit with higher priority (best cost to power ratio). The second is a time-dependent priority list in which we divide the pseudo-commitment decisions by the above pseudo cost over pseudo power ratio. A higher value indicates a unit more likely to be started.

Starting from our initial commitment guess \tilde{u} we first begin by computing the generation envelope, i.e., the minimum and maximum power the plants can generate over the whole time horizon at these commitment decisions. We now move from the first time step to the last one, if \tilde{D} is in the generation envelope, nothing more needs to be done. If generation is insufficient, we check if we can start the highest priority unit (if not done so already), we continue in this manner until generation covers load. If generation is in excess, we try to decommit the lowest priority unit (if not already off) and continue in this manner until the minimum generation is below load. The hence generated commitment decision is post-processed with an economic dispatch in order to finely adjust generation to actual load. We also post-process any of the generated commitment decisions u^j , $j = 1, \dots, p$ in order to retain the best one.

Recombining Heuristic. This method inspired from [48] recombines the earlier obtained primal iterates in order to find a feasible solution. For additional flexibility we add a slack variable, and therefore solve the following mixed-integer problem:

$$\begin{aligned} \min_{z \in \{0,1\}^{mp}, s \in \mathbb{R}_+^T} & \sum_{j=1}^p \sum_{i=1}^m c_i^j z_i^j + \sum_{t=1}^T c_t^{\text{imb}} s_t \\ \text{s.t.} & \sum_{j=1}^p z_i^j = 1, i = 1, \dots, m \quad \text{and} \quad \tilde{D} - \sum_{i=1}^m A^i \sum_{j=1}^p z_i^j x_i^j - s \leq s^u, \end{aligned}$$

where c^{imb} is a large imbalance related cost. The resulting optimal solution builds an initial commitment decision \tilde{u} , which is post-processed as explained previously. In order to keep the mixed-integer program small, we use the dual variables α and insert only elements of iteration j if α_j is sufficiently large, (e.g., $\alpha_j > 10^{-3}$).

References

1. L. Baccard, C. Lemaréchal, A. Renaud, and C. Sagastizábal. Bundle methods in stochastic optimal power management: a disaggregate approach using preconditioners. *Computation Optimization and Applications*, 20(3):227–244, 2001.
2. J. Batut and A. Renaud. Daily scheduling with transmission constraints: A new class of algorithms. *IEEE Transactions on Power Systems*, 7(3):982–989, 1992.
3. C. Beltran and F.J. Heredia. Unit commitment by augmented lagrangian relaxation: Testing two decomposition approaches. *Journal of Optimization Theory and Applications*, 112(2):295–314, 2002.
4. D. Bertsimas, E. Litvinov, X. A. Sun, J. Zhao, and T. Zheng. Adaptive robust optimization for the security constrained unit commitment problem. *IEEE Transactions on Power Systems*, 28(1):52–63, March 2013.
5. J. R. Birge and F. Louveaux. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384–392, 1988.
6. J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, New York, 1997.
7. A. Borghetti, A. Frangioni, F. Lacalandra, and C.A. Nucci. Lagrangian heuristics based on disaggregated bundle methods for hydrothermal unit commitment. *IEEE Transactions on Power Systems*, 18:313–323, 2003.
8. A. Borghetti, C. d’Ambrosio, A. Lodi, and S. Martello. Optimal scheduling of a multi-unit hydro power station in a short-term time horizon. Technical report, OR-12-13 University of Bologna, 2013.
9. A. Bruhns, G. Deurveilher, and J.S. Roy. A non-linear regression model for mid-term load forecasting and improvements in seasonality. *PSCC 2005 Luik*, 2005.
10. C.C. Carøe and R. Schultz. Dual decomposition in stochastic integer programming. *Operation Research Letters*, 24:37–45, 1999.
11. P. Carpentier, G. Cohen, J. C. Culioli, and A. Renaud. Stochastic optimization of unit commitment: a new decomposition framework. *IEEE Transactions on Power Systems*, 11(2), may 1996.
12. M. Carrión and J.M. Arroyo. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Transactions on Power Systems*, 21(3):1371–1378, 2006.
13. K. Cheung, D. Gade, C.S. Monroy, S. M. Ryan, J-P Watson, R.J-B. Wets, and D. L. Woodruff. Toward scalable stochastic unit commitment - part 2: Assessing solver performance. *Submitted (preprint available online)*, pages 1–8, 2014.
14. A. Daniilidis and C. Lemaréchal. On a primal-proximal heuristic in discrete optimization. *Math. Programming Series A*, 104:105–128, 2005.
15. W. de Oliveira and C. Sagastizábal. Level bundle methods for oracles with on demand accuracy. *Optimization Methods and Software*, 29(6):1180–1209, 2014.
16. D. Dentcheva and W. Römis. Optimal power generation under uncertainty via stochastic programming. In K. Marti and P. Kall, editors, *Stochastic Programming Methods and Technical Applications*, volume 458 of *Lecture Notes in Economics and Mathematical Systems*, pages 22–56. Springer Berlin Heidelberg, 1998. ISBN 978-3-540-63924-4.
17. D. Dentcheva and W. Römis. Duality gaps in nonconvex stochastic optimization. *Mathematical Programming*, 101(3):515–535, 200. doi: 10.1007/s10107-003-0496-1.
18. L. Dubost, R. Gonzalez, and C. Lemaréchal. A primal-proximal heuristic applied to french unitcommitment problem. *Mathematical programming*, 104(1):129–151, 2005.
19. J. E. Falk. Lagrange multipliers and nonconvex programs. *SIAM Journal on Control*, 7(4):534–545, July 1969. doi: 10.1137/0307039.
20. S. Feltenmark and K.C. Kiwiel. Dual applications of proximal bundle methods, including lagrangian relaxation of nonconvex problems. *SIAM Journal on Optimization*, 10(3):697–721, 2000.
21. Y. Feng, I. Rios, S. M. Ryan, K. Spürkel, J-P Watson, R.J-B. Wets, and D. L. Woodruff. Toward scalable stochastic unit commitment - part 1: Load scenario generation. *Submitted (preprint available online)*, pages 1–8, 2014.
22. E.C. Finardi and E.L. Da Silva. Solving the hydro unit commitment problem via dual decomposition and sequential quadratic programming. *IEEE Transactions on Power Systems*, 21(2):835–844, 2006.
23. A. Frangioni and C. Gentile. Solving non-linear single-unit commitment problems with ramping constraints. *Operations Research*, 54(4):767–775, 2006.
24. A. Frangioni, C. Gentile, and F. Lacalandra. Solving Unit Commitment Problems with General Ramp Constraints. *International Journal of Electrical Power and Energy Systems*, 30:316–326, 2008.
25. A. Frangioni, C. Gentile, and F. Lacalandra. Sequential Lagrangian-MILP Approaches for Unit Commitment Problems. *International Journal of Electrical Power and Energy Systems*, 33:585–593, 2011.
26. A. M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.
27. N. Gröwe-Kuska, K.C. Kiwiel, M.P. Nowak, W. Römis, and I. Wegner. Power management in a hydro-thermal system under uncertainty by lagrangian relaxation. In C. Greengard and A. Ruszczyński, editors, *Decision Making Under Uncertainty*, volume 128 of *The IMA Volumes in Mathematics and its Applications*, pages 39–70. Springer New York, 2002. ISBN 978-1-4419-3014-9.
28. J.B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms I*. Number 305 in Grundlehren der mathematischen Wissenschaften. Springer-Verlag, 2nd edition, 1996.
29. J.B. Hiriart-Urruty and C. Lemaréchal. *Convex Analysis and Minimization Algorithms II*. Number 306 in Grundlehren der mathematischen Wissenschaften. Springer-Verlag, 2nd edition, 1996.
30. N. Langrene, W. van Ackooij, and F. Bréant. Dynamic constraints for aggregated units: Formulation and application. *IEEE transactions on Power Systems*, 26(3):1349–1356, 2011.
31. C. Lemaréchal. *Lagrangian relaxation*. In [?] (Chapter 4). Springer-Verlag, 2001.
32. C. Lemaréchal and A. Renaud. A geometric study of duality gaps, with applications. *Math. Programming*, 90:399–427, 2001.
33. J. Malick, W. de Oliveira, and S. Zaourar. Nonsmooth optimization using uncontrolled inexact information. *Preprint: http://www.optimization-online.org/DB_FILE/2013/05/3892.pdf*, 2014.

34. A. Merlin, B. Lauzanne, J. Auge, and J. Ziglioli. Optimization of short-term scheduling of EDF hydraulic valleys with coupling constraints: the OVIDE model. In *Proc. PSCC Conf. 1981*, pages 345–354, 1981.
35. G. Morales-España, J.M. Latorre, and A. Ramos. Tight and compact MILP formulation for the thermal unit commitment problem. *IEEE Transactions on Power Systems*, 28(4):4897–4908, 2013.
36. G. Morales-España, J.M. Latorre, and A. Ramos. Tight and compact MILP formulation of start-up and shut-down ramping in unit commitment. *IEEE Transactions on Power Systems*, 28(2):1288–1296, 2013.
37. M.P. Nowak. *Stochastic Lagrangian relaxation in power scheduling of a hydrothermal system under uncertainty*. PhD thesis, Humboldt University Berlin, 2000.
38. M.P. Nowak and W. Römis. Stochastic lagrangian relaxation applied to power scheduling in a hydro-thermal system under uncertainty. *Annals Of Operations Research*, 100(1-4):251–272, December 2000.
39. R. Nürnberg and W. Römis. A two-stage planning model for power scheduling in a hydro-thermal system under uncertainty. *Optimization and Engineering*, 3:355–378, December 2003.
40. A.B. Philpott, M. Craddock, and H. Waterer. Hydro-electric unit commitment subject to uncertain demand. *European Journal of Operational Research*, 125:410–424, 2000.
41. R. T. Rockafellar and Roger J.-B. Wets. Scenarios and policy aggregation in optimization under uncertainty. *Mathematics of Operations Research*, 16(1):119–147, February 1991.
42. R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1st edition, 1970.
43. A. Ruszczyński. *Decomposition Methods (Chapter 3 in [?])*, volume 10 of *Handbooks in Operations Research and Management Science*. Elsevier, Amsterdam, 2003.
44. C. Sagastizábal. Divide to conquer: Decomposition methods for energy optimization. *Mathematical Programming*, 134(1):187–222, 2012.
45. H.D. Sherali and B.M.P. Fraticelli. A modification of benders’ decomposition algorithm for discrete subproblems: an approach for stochastic programs with integer recourse. *Journal of Global Optimization*, 22:319–342, 2002.
46. R.H. Shumway and D.S. Stoffer. *Time Series Analysis and Its Applications*. Springer Texts in Statistics. Springer, 5th edition, 2005.
47. M. Tahanan, W. van Ackooij, A. Frangioni, and F. Lacalandra. Large-scale unit commitment under uncertainty: a literature survey. *4OR*, available online, 2015. doi: 10.1007/s10288-014-0279-y.
48. S. Takriti and J.R. Birge. Using integer programming to refine lagrangian-based unit commitment solutions. *IEEE Transactions on Power Systems*, 15(1):151–156, 2000.
49. S. Takriti, J.R. Birge, and E. Long. A stochastic model for the unit commitment problem. *IEEE Transactions on Power Systems*, 11:1497–1508, 1996.
50. S. Takriti, B. Krasenbrink, and L.S.Y. Wu. Incorporating fuel constraints and electricity spot prices into the stochastic unit commitment problem. *Operations Research*, 48(2):268–280, April 2000.
51. W. van Ackooij and G. Doukopoulos. Méthodes de faisceaux pour la gestion de production. Technical Report H-R36-2012-02870-FR, EDF R&D, 2012.
52. W. van Ackooij, R. Henrion, A. Möller, and R. Zorgati. Joint chance constrained programming for hydro reservoir management. *Optimization and Engineering*, 15:509–531, 2014.
53. S.J. Wang, M. Shahidehpour, D.S. Kirschen, S. Mokhtari, and G.D. Irisarri. Short-term generation scheduling with transmission and environmental constraints using an augmented lagrangian relaxation. *IEEE Transactions on Power Systems*, 10(3):1294–1301, August 1995.
54. L. Wu, M. Shahidehpour, and T. Li. Stochastic security-constrained unit commitment. *IEEE Transactions on Power Systems*, 22(2), May 2007.
55. P. Xiong and P. Jirutitijaroen. Stochastic unit commitment using multi-cut decomposition algorithm with partial aggregation. In *IEEE Power and Energy Society General Meeting*, 2011.
56. H. Yan, P.B. Luh, and L. Zhang. Scheduling of hydrothermal power systems using the augmented lagrangian decomposition and coordination technique. In *American Control Conference 1994*, volume 2, pages 1558–1562, August 1994.
57. L. Zhao and B. Zeng. Robust unit commitment problem with demand response and wind energy. In *Proceedings of IEEE Power and Energy Society General Meeting, 2012*, 2012.
58. Q. Zheng, J. Wang, P. Pardalos, and Y. Guan. A decomposition approach to the two-stage stochastic unit commitment problem. *Annals of Operations Research*, 210(1):387–410, 2013.
59. F. Zhuang and F.D. Galiana. Towards a more rigorous and practical unit commitment by lagrangian relaxation. *IEEE Transactions on Power Systems*, 3(2):763–773, May 1988.